COMPUTERS & SECURITY 25 (2006) 435–444

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A secure extension of the Kwak–Moon group signcryption scheme☆

DongJin Kwak[a,*], SangJae Moon[b], Guilin Wang[c], Rorbert H. Deng[d]

[a]Next Generation Internet Division, KT Future Advanced Technology Laboratory, South Korea
[b]School of Electrical Eng. & Computer Science, Kyungpook National University, South Korea
[c]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
[d]School of Information Systems, Singapore Management University, Singapore 259756

ARTICLE INFO

Article history:
Received 4 April 2005
Revised 26 May 2006
Accepted 31 May 2006

Keywords:
Signature
Group signature
Signcryption
Public-key cryptography
Key agreement

ABSTRACT

This paper presents the secure extension of the Kwak–Moon group signcryption scheme [Kwak D, Moon S. Efficient distributed signcryption scheme as group signcryption. In: First applied cryptography and network security – ACNS'03. Lecturer notes in computer science, vol. 2846. Springer-Verlag; 2003. p. 403–17] as a countermeasure against the cryptanalysis in [Wang G, Deng RH, Kwak D, Moon S. Security analysis of two signcryption scheme. In: Information security conference – ISC 2004. Lecturer notes in computer science, vol. 3225. Springer-Verlag; 2004. p. 123–33]. The cryptanalysis revealed that the Kwak–Moon scheme cannot satisfy the properties of unforgeability, coalition-resistance, and traceability. Therefore, to avoid these weaknesses, while providing the same functions, we add confidentiality to the original group signature by distributing a shared secret among group members through an efficient group key agreement. However, in case of just combining a group signature and a group key agreement, if an attacker who does not belong to the group acquires a valid group signature, it is still possible for him to impersonate a valid group member and delegate the group. Thus, to avoid this possibility, the proposed scheme confirms whether or not the sender is equal to the signer by including a session key encryption in the signed message. In addition, we analyze the security of the proposed scheme and apply it to an anonymous statistical survey of attributes.

1. Introduction

1.1. Background

Signcryption (Zheng et al., 1997) is a new cryptographic method that can simultaneously provide message confidentiality and unforgeability with a lower computational and communicational overhead than the traditional signature-then-encryption. Following Zheng's pioneering work, a number of new schemes and improvements have been proposed, while literatures (An et al., 2002; Baek et al., 2002; Boyen, 2003; Steinfeld and Zheng, 2000) have studied the formal models and security proofs for such signcryption schemes. Originally, signcryption was created to send a message from a single sender to a designated receiver. In Zheng (1997), a variant scheme was proposed to support multiple designated receivers for the first time. Thereafter, Mu and Varadharajan (2000) proposed a distributed signcryption using the distributed encryption (Mu et al., 1999), in

☆This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

* *Corresponding author.*
E-mail address: djk@kt.co.kr (DongJin Kwak).

0167-4048/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2006.05.006

which any party can signcrypt a message and distribute it to a designated group, and any member in the receiving group can decrypt and verify the message using his unique decryption key each other. They also extended their scheme to a group application based on applying the group signature concept in Section E of Mu and Varadharajan (2000). In this extended scheme, a group manager generates and keeps the members' secret information. Yet, since this allows the possibility of forgery in the case of a dishonest group manager, exculpability is not satisfied, which is one of the main requirements for a group signature. Thus, to avoid this weakness and enhance the computational load, Kwak and Moon developed a concrete group signcryption scheme by modifying Mu et al.'s scheme.

1.2. *Our work*

In this paper, we recall a security analysis of the Kwak–Moon scheme (Wang et al., 2004) and propose a new encrypted group signature as a countermeasure against forgery of a membership certificate and signcryption itself in the scheme. Since the Kwak–Moon scheme does not satisfy the properties of unforgeability, coalition-resistance, and traceability, the proposed encrypted group signature solves the mentioned weakness of the Kwak–Moon scheme through effectively combining a group signature and a group key agreement. As such, the ACJT group signature (Ateniese et al., 2000) and the BCEP group key agreement (Bresson et al., 2003; Nam et al., 2004) are selected for the proposed scheme to deploy a concrete scheme. The ACJT group signature is currently considered to be a state of the art provably secure group signature and the BCEP is also a provably secure group key agreement that has very efficient computational loads. In addition, to prevent a malicious non-group member from using an acquired valid signature illegally, the senders sign a message together with a session key encryption value, thereby confirming whether or not the sender is the same as the signer. Finally, the security of the new encrypted group signature is analyzed and the proposed scheme can be applied to an anonymous statistical survey of attributes.

1.3. *Organization*

The next section presents the security of an encrypted group signature, then Section 3 briefly outlines the Kwak–Moon scheme and recalls its security. Section 4 explains the ACJT group signature and the BCEP group key agreement as components of the proposed scheme, then Section 5 presents the new encrypted group signature. Section 6 analyzes the security of the proposed scheme and compares the security properties with similar schemes such as the MV scheme and the Kwak–Moon scheme. Section 7 presents an application of the proposed scheme and some final conclusions are offered in Section 8.

2. Security of encrypted group signature

The proposed encrypted group signature are based on the strong RSA assumption (Barić and Pfitzmann, 1997; Fujisaki and Okamoto, 1997) and the DDH assumption (Boneh, 1998). Let ℓ_g be a suitable security parameter and $\mathcal{G}(\ell_g)$ denotes a set of groups whose order has the length ℓ_g and consists of two prime factors of length $(\ell_g - 2)/2$.

Assumption 1 (strong RSA assumption). *There exists a probabilistic polynomial-time algorithm $\mathcal{K}$ such that, for any probabilistic polynomial-time algorithm $\mathcal{A}$ and all sufficiently large ℓ_g, the probability that $\mathcal{A}$ on inputs G and z outputs $e \in \mathbb{Z} > 1$ and $u \in G$ satisfying $z = u^e \bmod n$ is negligible.*

Assumption 2 (decisional Diffie–Hellman assumption). *There exists a probabilistic polynomial-time algorithm $\mathcal{K}$ such that, for any probabilistic polynomial-time algorithm $\mathcal{A}$ and all sufficiently large ℓ_g, the probability that $\mathcal{A}$ on input g, g^x, g^y and $g^z \in_R G$ can distinguish whether g^{xy} and g^z are equal is negligible.*

A secure encrypted group signature should satisfy the following security requirements:

- *Correctness*: an encrypted signatured message produced by a group member must be accepted by the verification procedure.
- *Unforgeability*: only valid group members are able to sign a message on behalf of the group.
- *Anonymity*: with a valid decrypted message, identifying the individual who signed the message is computationally hard for anyone but the group manager.
- *Unlinkability*: determining whether two valid verified messages were generated by the same group member is computationally hard for anyone except the group manager.
- *Exculpability*: neither a group member nor the group manager can sign on behalf of other group members.
- *Traceability*: for any valid verified message, the group manager can open it and find the true signer.
- *Coalition-resistance*: a colluding subset of group members cannot generate a valid encrypted signature so that the group manager is unable to identify the colluding group members.
- *Confidentiality*: except for the members belonging to the receiving group, no other party can derive the decrypted signature message from the encrypted signature.
- *Membership identification*: only a group member can generate a valid encryption. Even if an attacker acquires a valid group signature, he cannot generate a valid encrypted group signature value.

3. Review of Kwak–Moon scheme

This section outlines the Kwak–Moon scheme and analyzes its security. Essentially, the Kwak–Moon scheme enables a member of a certain group to signcrypt a message on behalf of the group and sends it to a member in another group with anonymity.

3.1. *Procedure of the scheme*

The whole scheme consists of five stages: Setup, Join, Signcryption, Unsigncryption, and Open.

3.1.1. Setup

The GM (group manager) chooses a large prime p of the order q for $q|(p-1)$, $G=\langle g\rangle\subset\mathbb{Z}_p^*$ and $h\in_R\mathbb{Z}_p^*$. Then the GM selects an RSA modulus n, his RSA signature key d and a verification key e, where $e\cdot d=1 \bmod \phi(n)$. The GM publishes (p,q,g,h,n,e) and keeps d as his secret signature key.

3.1.2. Join

Each n entity i who wants to join a group generates his own secret key ϵ_i and computes the membership key $\tau_i=h^{\epsilon_i} \bmod p$. Next, he sends τ_i to the GM and proves to the GM that he knows the discrete logarithm of τ based on h by a zero-knowledge proof. Thereafter, the GM generates each membership certificate $\upsilon_i=\tau_i^d \bmod n$, then computes the coefficients of the following polynomial

$$f(x)=\prod_{i=1}^{n}(x-\tau_i)=\sum_{i=0}^{n}\alpha_i x^i$$

For user ℓ, define $A_\ell=\sum_{i=1}^{n-1}\sum_{j=1,i\neq j}^{n-1}\alpha_j\tau_\ell^i$, then another polynomial $F(\tau_\ell)$ has the following property:

$$F(\tau_\ell)=g^{f(\tau_\ell)}=g^{-A_\ell}g^{\alpha_0+\sum_{i=1}^{n-1}\alpha_i\cdot\sum_{i=1}^{n-1}\tau_\ell^i+\alpha_n\tau_\ell^n}=1 \bmod p$$

This is because $f(\tau_\ell)=0$ in $\mathbb{Z}_q$, where τ_ℓ is the membership key for user ℓ. To construct a group public key, the GM picks a $\gamma\in_R\mathbb{Z}_q$ and computes its inverse and $\rho_\ell=-\gamma A_\ell \bmod q$ for member ℓ. Now, the group public key is defined as a 4-tuple $\{\beta_0,\beta_1,\beta_2,\beta_3\}=\left\{g^{\alpha_0},g^{\sum_{i=1}^{n-1}\alpha_i},g^{\alpha_n},g^{\gamma^{-1}}\right\}$. As such, the GM keeps γ and all $\{\alpha_i\}$ secret and gives υ_ℓ and ρ_ℓ to his group member ℓ.

3.1.3. Signcryption

Assume that Alice with $(\epsilon_a,\tau_a,\upsilon_a)$ belongs to G_A and wants to signcrypt a message. As such, she sends it to the group G_B using group G_B's public key $\{\beta_0,\beta_1,\beta_2,\beta_3\}$. To signcrypt a message m, Alice does the following:

(1) Choose two numbers z, $t\in_R\mathbb{Z}_q$ and compute $k=g^z \bmod p=k_1\|k_2$.
(2) Compute $r=\mathcal{H}_{k_2}(m)$ and $s=z(r+\epsilon_a\cdot t)^{-1} \bmod q$.
(3) Compute $w=\mathcal{H}(m)$, $\lambda_a=(t^e\cdot\tau_a \bmod n)\bmod q$, $\delta_a=g^{\epsilon_a t} \bmod p$, and $\theta_a=t\cdot\upsilon_a \bmod n$.
(4) Compute $c_1\leftarrow\{a_0,a_1,a_2,a_3,a_4\}\leftarrow\left\{k\beta_0^{w\tau_a},\beta_1^{w\tau_a},\beta_2^{w\tau_a},\beta_3^{w\tau_a},g^{\lambda_a}\right\}$ and $c_2=E_{k_1}(\mathrm{ID}_{G_A}\|m\|r\|s\|\delta_a\|\theta_a)$

where $\mathcal{H}(\cdot)$ denotes a one-way hash function, $\mathcal{H}_k(\cdot)$ a keyed one-way hash function with key k, $E_k(\cdot)$ a symmetric key encryption with key k, and ID_{G_A} the identity of group G_A including (n,e).

3.1.4. Unsigncryption

Bob belonging to G_B can unsigncrypt the signcrypted message using his (τ_b,ρ_b) as follows:

(1) Discover the session key $k=a_0\cdot a_1^{\sum_{i=1}^{n-1}\tau_b^i}\cdot a_2^{\tau_b^n}\cdot a_3^{\rho_b}=g^z g^{f(\tau_b)w\tau_a}=g^z \bmod p$.
(2) Split k into k_1 and k_2.
(3) Decrypt c_2 with key k_1 and find $\mathrm{ID}_{G_A}\|m\|r\|s\|\delta_a\|\theta_a$.
(4) Compute $\lambda_a'=(\theta_a^e \bmod n)\bmod q$.
(5) Verify $r\stackrel{?}{=}\mathcal{H}_{k_2}(m)$, $k\stackrel{?}{=}(\delta_a\cdot g^r)^s$, and $a_4\stackrel{?}{=}g^{\lambda_a'}$.

3.1.5. Open

In the case of a dispute, Bob forwards the c_1, w, and his public key $\{\beta_0,\beta_1,\beta_2,\beta_3\}$ to the GM of G_A after decrypting c_2 and identifying the ID_{G_A}. The GM can then identify the group member, Alice, who issued the signcryption by testing whether a_i in c_1 is equal to $(\beta_i^w)^{\tau_\ell}$ for all his members' τ_ℓ. After this procedure, disputes can be solved by the GM.

3.2. Weakness of the scheme

In Kwak and Moon (2003), the scheme is claimed to satisfy all the required security properties for a group signature and signcryption. However, this subsection presents two universally forging attacks to show that the scheme is forgeable and not coalition-resistant.

3.2.1. Forging signcryption

To forge a group signcryption (r,s,δ_a,λ_a) needs to be found that satisfies the following verification equation:

$$g^z=(\delta_a\cdot g^r)^s \bmod p,\ \text{ and }\ \lambda_a=(t^e\cdot\tau_a \bmod n) \bmod q$$

First, we generate three random numbers $\bar{\epsilon}_i,\bar{t}$, and $\bar{z}\in_R\mathbb{Z}_q$ and compute $\bar{\tau}_i=h^{\bar{\epsilon}_i}\in\mathbb{Z}_p^*$ and $g^{\bar{z}} \bmod p=\bar{k}_1\|\bar{k}_2$. Then the following solution is evaluated for a message m:

$$r=\mathcal{H}_{\bar{k}_2}(m)$$
$$s=\bar{z}(r+\bar{\epsilon}_i\cdot\bar{t})^{-1} \bmod q$$
$$\delta_a=g^{\bar{\epsilon}_i\cdot\bar{t}} \bmod p$$
$$\lambda_a=(\bar{t}^e\cdot\bar{\tau}_i \bmod n)\bmod q$$

Thus, an attacker who generates $\bar{\epsilon}_i,\bar{t}$, and $\bar{z}\in_R\mathbb{Z}_q$ randomly and computes $\bar{\tau}_i=h^{\bar{\epsilon}_i}\in\mathbb{Z}_p^*$ and $g^{\bar{z}} \bmod p=\bar{k}_1\|\bar{k}_2$ can make a valid signcryption pair (c_1,c_2) verified by any target group members using the unsigncryption procedure. In case of a dispute, the target group member receiving the signcryption message forwards the part of signcryption message c_1, the hash value w and his public key $\{\beta_0,\beta_1,\beta_2,\beta_3\}$ to the GM of sending group after decrypting c_2 and identifying the sending group identity, ID_{G_A}. However, the GM cannot identify the group member who issued the signcryption by testing whether a_i in c_1 is equal to $(\beta_i^w)^{\tau_\ell}$ for all his member's τ_ℓ because $\bar{\tau}_i$ was not generated by his group members but generated by the attacker. Even though an attacker does not know membership certificate, the attack works simply.

3.2.2. Forged certificate by coalition

Each member's certificate υ_i can be forged by a coalition attack. Assume that there are more than two group member's certificates $\upsilon_1,\upsilon_2,\ldots$ and υ_i. If more than two group members collude to generate a new forged certificate, they can easily make a forged secret key ϵ_f and certificate υ_f easily as follows:

$$\upsilon_f=\upsilon_1\cdot\upsilon_2\cdots\upsilon_i=\left(h^{\sum_{j=1}^{i}\epsilon_j}\right)^d \bmod n$$
$$\epsilon_f=\sum_{j=1}^{i}\epsilon_j$$
$$\tau_f=h^{\epsilon_f} \bmod p$$

In the Kwak–Moon scheme, many forged pairs of (ϵ_f,υ_f) are valid membership certificates, implying that traceability can also not be satisfied with this scheme. Thus, a malicious attacker who generates a valid tuple $(\epsilon_f,\tau_f,\upsilon_f)$ by the coalition can make a valid signcryption verified by any target group

members using the unsigncryption procedure. In case of a dispute, the target group member receiving the signcryption message forwards the part of signcryption message c_1, the hash value w and his public key $\{\beta_0,\beta_1,\beta_2,\beta_3\}$ to the GM of sending group after decrypting c_2 and identifying the sending group identity, ID_{G_A}. However, The GM cannot identify the group member who issued the signcryption by testing whether a_i in c_1 is equal to $(\beta_i^w)^{\tau_\ell}$ for all his member's τ_ℓ because τ_f was generated by malicious group members using coalition.

4. Preliminaries

This section briefly describes the two preliminaries used in the following proposed scheme.

4.1. *The ACJT group signature scheme*

This gives an overview of the ACJT group signature scheme, however, readers familiar with ACJT may skip this section with no loss of continuity. In its interactive, identity escrow form, the ACJT scheme has been proven to be secure and coalition-resistant under the strong RSA and DDH assumptions. In addition, the security of the non-interactive group signature scheme relies additionally on the Fiat–Shamir heuristic, also known as the random oracle model (Bellare and Rogaway, 1993).

Let ϵ, k, ℓ_p be the security parameters and let λ_1, λ_2, γ_1, and γ_2 denote lengths satisfying $\lambda_1 > \epsilon(\lambda_2 + k) + 2$, $\lambda_2 > 4\ell_p$, $\gamma_1 > \epsilon(\gamma_2 + k) + 2$, and $\gamma_2 > \lambda_1 + 2$. Define the integral ranges $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$. The ACJT scheme is then defined by the set of quadratic residues QR(n) where $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ (p,q,p',q' are all prime numbers). The group public key is $\mathcal{Y} = (n, a, a_0, y = g^x, g, h)$ where a, a_0, g and h are randomly selected in QR(n) and y is the GM (Group Manager)'s public key. The corresponding secret key known only to the GM is $\mathcal{S} = (p', q', x)$.

To join the group, a user engages in a JOIN protocol with the GM and receives a membership certificate $[A_i,e_i]$, where $A_i = (a^{x_i} a_0)^{1/e_i} \bmod n$ with a prime $e_i \in \Gamma$ and $x_i \in \Lambda$ (x_i is known only to the user). Therefore (x_i,A_i,e_i) is P_i's group signing key.

4.1.1. *Sign*

To anonymously sign a message m, a group member with a group signing key (x_i,A_i,e_i) has to prove possession of the signing key without revealing it by an SPK (signature based on proof of knowledge) (Camenisch and Stadler, 1997; Camenisch and Michels, 1998a,b). In particular, a group member P_i first computes:

$$T_1 = A_i y^\omega \bmod n,\ \ T_2 = g^\omega \bmod n,\ \ T_3 = g^{e_i} h^\omega \bmod n$$

where $\omega \in_R \{0,1\}^{2\ell_p}$. Then to show that he prepared (T_1,T_2,T_3) correctly, he generates the following SPK:

$$\mathrm{SPK}\Big\{(\alpha,\beta,\gamma,\delta) : a_0 = T_1^\alpha / a^\beta y^\gamma \wedge 1 = T_2^\alpha / g^\gamma \wedge T_2 = g^\delta \wedge T_3 = g^\alpha h^\delta \wedge \alpha \in \Gamma \wedge \beta \in \Lambda\Big\}(m)$$

The SPK represents a signature of knowledge of:

- a value x_i such that $(a^{x_i} a_0)^{1/e_i}$ is the value that is ElGamal-encrypted in (T_1,T_2) under the GM's public key y.
- an e_i-th root of that encrypted value, where e_i is the first part of the representation of T_3 w.r.t. g and h, such that e_i lies in Γ.

Therefore, for a given hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ and a security parameter $e > 1$ that controls the tightness of the statistical zero-knowledge, the signer P_i does the following to sign the message m:

(1) Choose $\omega \in_R \{0,1\}^{2\ell_p}$ and compute:

$$T_1 = A_i y^\omega \bmod n,\ \ T_2 = g^\omega \bmod n,\ \ T_3 = g^{e_i} h^\omega \bmod n$$

(2) Choose four random numbers r_1, r_2, r_3, and r_4, such that $r_1 \in_R \pm\{0,1\}^{\epsilon(\gamma_2+k)}$, $r_2 \in_R \pm\{0,1\}^{\epsilon(\lambda_2+k)}$, $r_3 \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\ell_p+k+1)}$, and $r_4 \in_R \pm\{0,1\}^{\epsilon(2\ell_p+k)}$.

(3) Compute the following four values of d_1, d_2, d_3, and d_4 (all in $\mathbb{Z}_n$):

$$d_1 = T_1^{r_1}/(a^{r_2} y^{r_3}) \bmod n,\ \ d_2 = T_2^{r_1}/g^{r_3} \bmod n$$
$$d_3 = g^{r_4} \bmod n,\ \ d_4 = g^{r_1} h^{r_4} \bmod n$$

(4) Evaluate the hash value $c = \mathcal{H}(g\|h\|y\|a_0\|a\|T_1\|T_2\|T_3\|d_1\|d_2\|d_3\|d_4\|m)$

(5) Calculate the following four numbers s_1, s_2, s_3, and s_4 (all in $\mathbb{Z}$):

$$s_1 = r_1 - c(e_i - 2^{\gamma_1}),\ \ s_2 = r_2 - c(x_i - 2^{\lambda_1})$$
$$s_3 = r_3 - c e_i \omega,\ \ s_4 = r_4 - c\omega$$

(6) Output signature $(c,s_1,s_2,s_3,s_4,T_1,T_2,T_3)$ of message m.

4.1.2. *Verify*

A verifier can check the validity of a group signature $(c,s_1,s_2,s_3,s_4,T_1,T_2,T_3)$ of the message m as follows:

(1) Check whether $c \in \{0,1\}^k$, and $s_1 \in \pm\{0,1\}^{\epsilon(\gamma_2+k)+1}$, $s_2 \in \pm\{0,1\}^{\epsilon(\lambda_2+k)+1}$, $s_3 \in \pm\{0,1\}^{\epsilon(\lambda_1+2\ell_p+k+1)+1}$, $s_4 \in \pm\{0,1\}^{\epsilon(2\ell_p+k)+1}$ and T_1, T_2, $T_3 \in \mathbb{Z}_n$.

(2) Accept the signature if and only if $c \equiv \mathcal{H}(g\|h\|y\|a_0\|a\|T_1\|T_2\|T_3\|d_1'\|d_2'\|d_3'\|d_4'\|m)$,

where d_1', d_2', d_3', d_4' are computed by the following equations:

$$d_1' = a_0^c T_1^{s_1 - c2^{\gamma_1}} / \left(a^{s_2 - c2^{\lambda_1}} y^{s_3}\right) \bmod n,\ \ d_2' = T_2^{s_1 - c2^{\gamma_1}} / g^{s_3} \bmod n$$
$$d_3' = T_2^c g^{s_4} \bmod n,\ \ d_4' = T_3^c g^{s_1 - c2^{\gamma_1}} h^{s_4} \bmod n$$

In the case the actual signer has to be subsequently identified, the GM opens the ElGamal encryption to reveal the group certificate A_i, unique to each member. In addition, the GM provides proof that:

$$D \log_g y = D \log_{T_2}(T_1/A_i)$$

which demonstrates that the encryption was opened correctly.

4.2. *The BCEP group key agreement scheme*

The BCEP scheme is a very efficient and provably secure group key agreement based on the security of the underlying signature used to authenticate messages and on the CDH (computational Diffie–Hellman) problem.

1. First, each client (user) U_i in group $\mathcal{G}_c$ chooses a random $\hat{x}_i \in \mathbb{Z}_q^*$ and computes $\hat{y}_i = \hat{g}^{\hat{x}_i}$, where $\hat{g}$ is a generator of a prime order q in a finite cyclic group $\mathbb{G} = \langle \hat{g} \rangle$. Then the

client signs $\hat{y}_i$ to obtain signature σ_i and sends $(\hat{y}_i, \sigma_i)$ to the server $\mathcal{SR}$, who chooses a random $\hat{x}_s \in \mathbb{Z}_q^*$ and computes $\hat{y}_s = \hat{g}^{\hat{x}_s}$.

2. For each client U_i, the server $\mathcal{SR}$ verifies the signature σ_i and computes $\alpha_i = \hat{y}_i^{\hat{x}_s}$. Then the server initializes the counter c to 0, computes the shared secret value

$$\kappa = \mathcal{H}_0(c\|\alpha_1\|\ldots\|\alpha_n)$$

and compute $\kappa_i = \kappa \oplus \mathcal{H}_1(c\|\alpha_i)$, where n is the number of clients, $\mathcal{H}_0$ denotes a one-way hash function with length ℓ_0, and $\mathcal{H}_1$ is another one-way hash function with length ℓ_1 that need not be equal to ℓ_0. The server signs the message $c\|\kappa_i\|\hat{y}_s$ to obtain signature σ_s and sends $(c, \kappa_i, \hat{y}_s, \sigma_s)$ to each user.

3. Each client U_i verifies the signature σ_s and computes $\alpha_i = \hat{y}_s^{\hat{x}_s}$ then recovers the shared secret value κ and the session key sk as described below:

$$\kappa = \kappa_i \oplus \mathcal{H}_1(c\|\alpha_i) \text{ and } sk = \mathcal{H}_2(\kappa\|\mathcal{G}_c\|\mathcal{SR})$$

where $\mathcal{H}_2$ is the other one-way hash function with length ℓ that need not be equal to ℓ_0, ℓ_1.

5. The proposed scheme

This section presents the new encrypted group signature scheme as a countermeasure against the attack of the Kwak–Moon scheme. As such, the proposed scheme solves the above-mentioned weakness of the Kwak–Moon scheme, like forged certificates and forged signatures, by applying the ACJT group signature. Plus, to effectively share the session key between the sender (signer) and the receiver (verifier), the BCEP group key agreement based on the strong RSA assumption is also included. The whole scheme consists of five steps: Setup, Join, Encrypted signature, Verification and Open. The Setup and Open procedures are the same as in the ACJT scheme.

5.1. *Setup*

The GM does the following:

- Chooses two random secret ℓ_p-bit primes p', q' such that $p = 2p' + 1$, $q = 2q' + 1$ are prime. Set the modulus $n = pq$.
- Chooses random elements a, a_0, g, $h \in_R QR(n)$ (of order $p'q'$).
- Chooses his secret element $x \in_R \mathbb{Z}_{p'q'}$ and sets $y = g^x \bmod n$.

Then the group public key is $\mathcal{Y} = (n, a, a_0, y, g, h)$ and the corresponding secret key is $\mathcal{S} = (p', q', x)$.

5.2. *Join*

Here, the BCEP group key agreement scheme is introduced for distributing additional secret information to each member. The original security of the BCEP scheme is based on DLP. However, in the proposed scheme, the BCEP scheme is based on the strong RSA assumption to be compatible with the underlying group signature. The detailed Join steps are as follows:

- User U_i who wants to join a group generates his own secret x_i and his certificate (A_i, e_i) with his group manager through interactive protocol as in the ACJT scheme, where A_i denotes $(a^{x_i} a_0)^{1/e_i} \bmod n$.
- U_i computes $y_i = g^{x_i} \bmod n$ and sends y_i and the signature σ_i for y_i to the GM who chooses a random $x \in_R \mathbb{Z}_{p'q'}^*$ and computes $y = g^x \bmod n$.
- After receiving each member's (y_i, σ_i), the GM checks whether (y_i, σ_i)s are valid or not. If they are all correct, the GM then computes the members' $\alpha_i = y_i^x \bmod n$ and generates the group secret shared key $\kappa = \mathcal{H}_0(c\|\alpha_1\|\ldots\alpha_N)$, where c is the IV (Initial Vector), $\mathcal{H}_0(\cdot)$ is a one-way hash function, and N is the size of the group.
- The GM sends $(\kappa_i, c, y, \sigma_s)$ to each member, where κ_i is $\kappa \oplus \mathcal{H}_1(c\|\alpha_i)$, $\mathcal{H}_1(\cdot)$ is another one-way hash function different from $\mathcal{H}_0(\cdot)$, and σ_s denotes the signature of $\kappa_i\|c\|y$. Each member then checks the signature σ_s then computes $\alpha_i = y^{x_i} \bmod n$ and recovers the shared secret value κ. As such, the GM can share the group secret key κ with all the group members and publish $\Omega = g^k \bmod n$.

5.3. *Encrypted signature*

Consider two designated groups G_a and G_b and assume that U_i belonging to group G_a wishes to send an encrypted signature message to group G_b on the behalf of group G_a and that V_j is one of the recipients belonging to group G_b. The G_a's public key is $\mathcal{Y}_a = (n, a, a_0, y, g, h, \Omega_a)$ and the corresponding GM's secret key is $\mathcal{S}_a = (p', q', x)$, while G_b's public key is $\mathcal{Y}_b = (\overline{n}, \overline{a}, \overline{a}_0, \overline{y}, \overline{g}, \overline{h}, \Omega_b)$ and the corresponding GM's secret key is $\mathcal{S}_b = (\overline{p'}, \overline{q'}, \overline{x})$.

Armed with $(x_i, A_i, e_i, y_i, \alpha_i, \Omega_a)$, a group member U_i in group G_a can then generate and send an encrypted group signature of message (C_1, C_2) to one member, V_j, in group G_b as follows:

- Choose $r_1, r_2 \in_R \{0,1\}^{2\ell_p}$ and compute the session key $k_s = F(g^{r_2})$, where $F(\cdot)$ is a suitable hash function such that $|F(\cdot)| < |\overline{n}|$.
- Compute $C_1 \leftarrow (b_0, b_1) \leftarrow (\overline{g}^{r_1}, k_s \Omega_b^{r_1})$ and $C_2 = E_{k_s}(m\|\sigma(m\|C_1)\|ID_{G_a})$

  where $\sigma(\cdot)$ is ACJT group signature, $E_k(\cdot)$ denotes a symmetric encryption with key k, and ID_{G_a} is the identity of G_a.

- Send (C_1, C_2) to a member of group G_b.

5.4. *Verification*

User V_j, one of the receiver group G_b, uses his group's κ_b to do the following with ciphertext (C_1, C_2):

- Recover the session key $k_s \leftarrow b_1/(b_0)^{\kappa_b}$.
- Decrypt $D_{k_s}(C_2) = m\|\sigma(m\|C_1)\|ID_{G_a}$ with the session key k_s, where $D_k(\cdot)$ denotes a symmetric decryption with key k.
- Verify the signature using the same verification procedure as in the ACJT scheme.

It is obvious that the recipient can verify an encrypted signature using the above process. The only way to decrypt the encrypted signature (C_1, C_2) is to have the shared group

key κ_b for ElGamal-type decryption. After decrypting, V_j first concatenates the message m and C_1 then verifies the ACJT group signature with the message $m\|C_1$.

5.5. *Open*

In the case of a dispute, the GM can decrypt (T_1,T_2) in a decrypted C_2 message to find the membership certificate A_i as follows:

- Check the signature's validity via the verification procedure.
- Recover A_i as $A_i = T_1/T_2^x$.
- Prove that $\log_g y = \log_{T_2}(T_1/A_i \bmod n)$.

6. Security of proposed scheme

This section presents a security analysis of the proposed scheme and compares the security properties with those of the MV scheme (Mu and Varadharajan, 2000), the Kwak–Moon scheme, and the proposed scheme, as shown in Table 1.

As an encrypted group signature, the proposed scheme should have both group signature's security properties and confidentiality at the same time. In addition, it should also have a membership identification that confirms whether or not the signer of the signature is equal to the sender to guard against possible misuse by a non-group member with another member's group signature. For the proposed encrypted group signature, most of these properties depend on the ACJT scheme that is based on the strong RSA assumption.

6.1. *Correctness*

The correctness of the proposed scheme can be proved by inspection of its verification procedure.

- First, the encrypted signature (C_1,C_2) can be decrypted by the group members who have the right-shared key κ_b for ElGamal-type decryption as follows:

$$k_s \leftarrow b_1/(b_0)^{\kappa_b} = k_s\Omega_b^{r_1}/(\overline{g}^{r_1})^{\kappa_b} = k_s(\overline{g}^{\kappa_b})^{r_1}/(\overline{g}^{r_1})^{\kappa_b},$$
$$D_{k_s}(C_2) = m\|\sigma(m\|C_1)\|\mathrm{ID}_{G_A}.$$

- After decrypting the ciphertext, a verifier can check the validity of the signature $\sigma(m\|C_1) = (c,s_1,s_2,s_3,s_4,T_1,T_2,T_3)$ of the message $m' = (m\|C_1)$ through verifying

$$\mathrm{SPK}\Big\{(\alpha,\beta,\gamma,\delta) : a_0 = T_1^{\alpha}/a^{\beta}y^{\gamma} \wedge 1 = T_2^{\alpha}/g^{\gamma} \wedge T_2 = g^{\delta} \wedge T_3 = g^{\alpha}h^{\delta} \wedge \alpha \in \Gamma \wedge \beta \in \Lambda\Big\}(m')$$

where $\alpha = e_i$, $\beta = x_i$, $\gamma = \omega e_i$, and $\delta = \omega \in_R \{0,1\}^{2\ell_p}$, as follows:

$$c \stackrel{?}{=} \mathcal{H}(g\|h\|a_0\|a\|T_1\|T_2\|T_3\|d_1'\|d_2'\|d_3'\|d_4')$$

where d_1', d_2', d_3', d_4' are computed like the below equation:

$$d_1' = a_0^c T_1^{s_1 - c2^{\gamma_1}}/(a^{s_2 - c2^{\lambda_1}} y^{s_3}) \bmod n, \quad d_2' = T_2^{s_1 - c2^{\gamma_1}}/g^{s_3} \bmod n,$$
$$d_3' = T_2^c g^{s_4} \bmod n, \quad d_4' = T_3^c g^{s_1 - c2^{\gamma_1}} h^{s_4} \bmod n.$$

As such, only a valid encrypted group signature will be accepted by the verification step.

6.2. *Anonymity*

All information about the group identity and entity identity is encrypted by symmetric encryption $E_k(\cdot)$ in C_2. Thus, no one except the designated receiving group members can decrypt the signature. Even though the receivers can decrypt it, only the GM knows information about the signer. Because deciding whether some group member with certificate (A_i,e_i) originated requires deciding whether the three discrete logarithms $\log_y T_1/A_i$, $\log_g T_2$, and $\log_h T_3/g^{e_i}$ are equal, this is assumed to be infeasible under the decisional Diffie–Hellman assumption and hence anonymity is guaranteed. It means that no one can find out any information on the identity of the signer.

6.3. *Unforgeability*

It is not feasible to forge the signature and certificate using the mentioned forgery in Section 2. Each user has a different membership certificate (A_i,e_i) and SPK when using his certificate under the ACJT scheme, thus it is impossible to forge the certificate and signature. As such, only valid group members are able to sign a message on behalf of the group. This is an immediate consequence of Lemma 1.

Lemma 1. *Under the strong RSA assumption, the interactive protocol underlying the ACJT group signature scheme is a statistical zero-knowledge proof of knowledge of a membership certificate and a corresponding membership secret key.*

6.4. *Unlinkability*

Deciding whether or not two valid signatures (C_1,C_2) and $(\overline{C}_1,\overline{C}_2)$ were computed by the same group and same group member is computationally hard to figure out. C_1 is computed using a randomly chosen value r_1, while C_2 is the encrypted value using a randomly chosen session key. Thus, even though the receivers decrypt C_2, they cannot determine any association due to the unlinkability of the ACJT scheme.

6.5. *Confidentiality*

The whole signed message (C_1,C_2) is encrypted by symmetric and asymmetric encryption. Plus, group ID confidentiality is also satisfied, since the group ID is included in the encrypted

Table 1 – Security comparison of MV scheme, Kwak–Moon scheme, and the proposed scheme

Security property	MV scheme	Kwak–Moon scheme	Proposed scheme
Correctness	Y	Y	Y
Anonymity	Y	Y	Y
Unlikability	Y	Y	Y
Confidentiality	Y	Y	Y
Exculpability	N	Y	Y
Traceability	Y	N	Y
Unforgeability	Y	N	Y
Coalition-resistance	Y	N	Y

Y, satisfied; N, not satisfied.

message, C_2. Thus, an adversary cannot discover any information without finding out the session key k_s with C_1. Finding out the key is based on CDH (Computational Diffie–Hellman) problem like ElGamal encryption scheme. Since there is no sender ID information and public parameter in the sign-and-encryption message in our scheme, it is more difficult to break than CDH problem. If an attacker is able to involve the group secret key agreement step and obtain group shared secret key κ, he can decrypt the message. In this case, the adversary's advantage $\mathrm{Adv}(\mathcal{A})$ for finding out group shared secret key κ is denoted by

$$\mathrm{Adv}(\mathcal{A}) \leq 2 \cdot \mathrm{Succ}_{\mathrm{SIGN}}^{\mathrm{cma}}(t, q_s) + 2q_s q_h \cdot \mathrm{Succ}_{\mathbb{G}}^{\mathrm{cdh}}(t)$$

where $\mathrm{Succ}_{\mathrm{SIGN}}^{\mathrm{cma}}(t, q_s)$ denotes the probability that an oracle produces a new and valid (message, signature) pair with t working time and q_s number of signing oracle queries, $\mathrm{Succ}_{\mathbb{G}}^{\mathrm{cdh}}(t)$ means a probability of solving CDH problem with t working time, and q_h is the number of query to the hash oracles ($\mathcal{H}_0$ and $\mathcal{H}_1$). It shows that the security of the group key agreement is based on CDH and on adaptive chosen message attack (CMA) of the signature. So, it is infeasible for attackers to find out the shared key κ with illegal and illegitimate methods.

6.6. *Membership identification*

Senders sign the message with a session key encryption C_1. Thus, even if the signature $\sigma(m\|C_1)$ is revealed to a non-group members, the verifiable session key encryption C_1 cannot be computed from $\sigma(m\|C_1)$. Therefore, only valid group members can generate valid encrypted group signatures.

6.7. *Traceability*

In a dispute, the GM can decrypt (T_1,T_2) in a decrypted C_2 message to find the membership certificate A_i as follows:

- Recover A_i as $A_i = T_1/T_2^x$.
- Prove $\log_g y = \log_{T_2}(T_1/A_i \bmod n)$.

6.8. *Coalition-resistance*

The underlying group signature scheme, ACJT, is a provably secure coalition-resistance scheme by Lemma 2 and the secret shared key κ computed by the GM cannot be computed by any coalition attack except valid group members who knows his $\alpha_i = y^{x_i}$, where x_i is entity's PKC (Public Key Cryptosystem) private key. The only way to find the session key is to find the x satisfying $\Omega = g^x \bmod n$. It is also based on strong RSA problem. As such, the proposed scheme has the coalition-resistance property.

Lemma 2. *Under the strong RSA assumption, a group certificate* $[A_i = (a^{x_i}a_0)^{1/e_i} \bmod n, e_i]$ *with* $x_i \in \Lambda$ *and* $e_i \in \Gamma$ *can be generated only by the group manager provided that the number K of certificates the group manager issues is polynomially bounded.*

6.9. *Exculpability*

The GM cannot obtain any information about a user's secret key x_i apart from a^{x_i} based on DDH and the strong RSA assumption. It is completely different from the existing group signcryption scheme. Plus, the ACJT signature scheme includes an unconditionally binding commitment to the membership certificate (e_i,A_i). Hence, neither a group member nor the group manager can sign on behalf of other group members.

7. Application

In this section, the proposed scheme is applied to an anonymous statistical survey of attributes, which was previously proposed in Nakanishi and Sugiyama (2001) and Nakanishi et al. (2002). The purpose of the system is to allow a service provider to collect personal information attributes such as gender, age, job and the like from a user. This information is useful for marketing purpose. On the other hand, users desire to use the service anonymously, since disclosing their personal information may enable the distributor to discover their identity. This system resolves the problem by using some TTPs'. This allows the distributor to obtain only statistics about the attributes without obtaining the individual identity and individual attributes themselves. The main requirements of this system are both the correctness of the statistics and the anonymity of users.

The participants in this system are the distributor, users, trustees, and an attribute authority. It is assumed that the attribute authority can be convinced of the users' genuine attributes, and that the authority issues a correct certificate of the attributes instead of the identity. Here, an anonymous statistical survey system can be easily constructed, where the property of group ID anonymity is used, like Nakanishi et al. (2002). The sender group ID is encrypted in C_2, so only designated group members can know the group ID. The details are as follows:

- *Setup*: the encrypted group signature is set up, where the attribute authority plays the role of the GM.
- *Registration*: to join the system, a user conducts the join protocol of the encrypted group signature with the attribute authority, where the user joins the group based on a corresponding attribute value.
- *Offer*: during the service, the user sends their encrypted group signature to the distributor for decryption by a certain trustee. Users select one designated trustee who is recommended by the distributor and let the distributor know which trustee is designated.
- *Generate*: the distributor gives the trustees the collected signatures. The trustees decrypt and verify the signatures then reveal the groups. The revealed groups indicate the statistics of the attributes.

Owing to the confidentiality, anonymity, and unlinkability of the encrypted group signature, the distributor obtains no information beyond the statistics. The correctness of the statistics is also assured due to the unforgeability of the encrypted group signature.

8. Conclusion

Security flaws in the Kwak–Moon scheme were briefly recalled, including certificate and signcryption forgery by an

invalid attacker. Thus, to avoid such attacks, while providing the same function as the Kwak–Moon scheme, a new encrypted group signature was proposed as a countermeasure. The proposed scheme is based on the original group signature, plus a simple additional computation for supporting confidentiality. The new scheme has potential applications in electronic commerce and other areas, such as an anonymous statistical survey of attributes.

Acknowledgement

We would like to thank anonymous reviewers for their helpful comments to improve our manuscript.

Appendix

We provide here the proofs of the lemmas stated in the paper. The proofs are related to proofs in Ateniese et al. (2000).

Lemma 1. *Under the strong RSA assumption, the interactive protocol underlying the ACJT group signature scheme is a statistical zero-knowledge proof of knowledge of a membership certificate and a corresponding membership secret key*.

Proof. We have to show that the knowledge extractor is able to solve the strong RSA problem and recover the group certificate and corresponding membership secret once it has found two accepting tuples $(T_1,T_2,T_3,d_1,d_2,d_3,d_4,c,s_1,s_2,s_3,s_4)$ and $(T_1,T_2,T_3,d_1,d_2,d_3,d_4,\tilde{c},\tilde{s}_1,\tilde{s}_2,\tilde{s}_3,\tilde{s}_4)$.

First, we show that the extractor is able to solve the strong RSA problem when obtains the two accepting tuples.

- Since $d_3 \equiv g^{s_4}T_2^{c} \equiv g^{\tilde{s}_4}T_2^{\tilde{c}} (\bmod n)$, it follows that $g^{s_4-\tilde{s}_4} \equiv T_2^{\tilde{c}-c} (\bmod n)$. Letting $\delta_4 = \gcd(s_4-\tilde{s}_4, \tilde{c}-c)$, by the extended Euclidean algorithm, there exist $\alpha_4, \beta_4 \in \mathbb{Z}$ s.t. $\alpha_4(s_4-\tilde{s}_4)+\beta_4(\tilde{c}-c)=\delta_4$. Hence,

$$g \equiv g^{(\alpha_4(s_4-\tilde{s}_4)+\beta_4(\tilde{c}-c))/\delta_4}$$
$$\equiv ((g^{((s_4-\tilde{s}_4)/(\tilde{c}-c))})^{\alpha_4} g^{\beta_4})^{((\tilde{c}-c)/\delta_4)}$$
$$\equiv (T_2^{\alpha_4} g^{\beta_4})^{((\tilde{c}-c)/\delta_4)} (\bmod n).$$

Since a $((\tilde{c}-c)/\delta_4)$th root of g can be computed as $T_2^{\alpha_4} g^{\beta_4}$, this implies it is a contradiction of the strong RSA assumption.

- Since $d_4 \equiv g^{s_1}h^{s_4}(T_3 g^{-2^{\gamma_1}})^{c} \equiv g^{\tilde{s}_1}h^{\tilde{s}_4}(T_3 g^{-2^{\gamma_1}})^{\tilde{c}} (\bmod n)$, it follows that $g^{s_1-\tilde{s}_1} \equiv (h^{-((s_4-\tilde{s}_4)/(\tilde{c}-c))} \; T_3 g^{-2^{\gamma_1}})^{\tilde{c}-c} (\bmod n)$. Let $\delta_1 = \gcd(s_1-\tilde{s}_1, \tilde{c}-c)$. By the extended Euclidean algorithm, there exist $\alpha_1, \beta_1 \in \mathbb{Z}$ s.t. $\alpha_1(s_1-\tilde{s}_1)+\beta_1(\tilde{c}-c)=\delta_1$. Therefore,

$$g \equiv g^{(\alpha_1(s_1-\tilde{s}_1)+\beta_1(\tilde{c}-c))/\delta_1}$$
$$\equiv \left[\left(T_3 g^{-2^{\gamma_1}} h^{-((s_4-\tilde{s}_4)/(\tilde{c}-c))}\right)^{\alpha_1} g^{\beta_1}\right]^{(\tilde{c}-c/\delta_1)} (\bmod n).$$

Since a $((\tilde{c}-c)/\delta_1)$th root of g can be computed as $(T_3 g^{-2^{\gamma_1}} h^{-((s_4-\tilde{s}_4)/(\tilde{c}-c))})^{\alpha_1} g^{\beta_1}$, this implies it is a contradiction of the strong RSA assumption.

- Since $d_2 \equiv T_2^{s_1} g^{-s_3}(T_2^{-2^{\gamma_1}})^{c} \equiv T_2^{\tilde{s}_1} g^{-\tilde{s}_3}(T_2^{-2^{\gamma_1}})^{\tilde{c}} (\bmod n)$, it follows that $g^{s_3-\tilde{s}_3} \equiv (T_2^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}})^{\tilde{c}-c} (\bmod n)$. Let $\delta_3 = \gcd(s_3-\tilde{s}_3, \tilde{c}-c)$. By the extended Euclidean algorithm, there exist $\alpha_3, \beta_3 \in \mathbb{Z}$ s.t. $\alpha_3(s_3-\tilde{s}_3)+\beta_3(\tilde{c}-c)=\delta_3$. Therefore,

$$g \equiv g^{(\alpha_3(s_3-\tilde{s}_3)+\beta_3(\tilde{c}-c))/\delta_3}$$
$$\equiv \left[\left(T_2^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}}\right)^{\alpha_3} g^{\beta_3}\right]^{((\tilde{c}-c)/\delta_3)} (\bmod n).$$

Since a $((\tilde{c}-c)/\delta_3)$th root of g can be computed as $(T_2^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}})^{\alpha_3} g^{\beta_3}$, this implies it is a contradiction of the strong RSA assumption.

- Since $d_1 \equiv T_1^{s_1} a^{-s_2} y^{-s_3} (T_1^{-2^{\gamma_1}} a^{2^{\lambda_1}} a_0)^{c} \equiv T_1^{\tilde{s}_1} a^{-\tilde{s}_2} y^{-\tilde{s}_3} (T_1^{-2^{\gamma_1}} a^{2^{\lambda_1}} a_0)^{\tilde{c}} (\bmod n)$, we obtain $a^{s_2-\tilde{s}_2} \equiv (T_1^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}} y^{-((s_3-\tilde{s}_3)/(\tilde{c}-c))} a^{-2^{\lambda_1}} a_0^{-1})^{\tilde{c}-c} (\bmod n)$. Let $\delta_2 = \gcd(s_2-\tilde{s}_2, \tilde{c}-c)$. By the extended Euclidean algorithm, there exist $\alpha_2, \beta_2 \in \mathbb{Z}$ s.t. $\alpha_2(s_2-\tilde{s}_2)+\beta_2(\tilde{c}-c)=\delta_2$. Therefore,

$$a_0 \equiv a_0^{(\alpha_2(s_2-\tilde{s}_2)+\beta_2(\tilde{c}-c))/\delta_2}$$
$$\equiv \left[\left(T_1^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}} y^{-((s_3-\tilde{s}_3)/(\tilde{c}-c))} a^{-2^{\lambda_1}} a_0^{-1}\right)^{\alpha_2} a_0^{\beta_2}\right]^{((\tilde{c}-c)/\delta_2)} (\bmod n).$$

Since a $((\tilde{c}-c)/\delta_2)$th root of a_0 can be computed as $(T_1^{((s_1-\tilde{s}_1)/(\tilde{c}-c))+2^{\gamma_1}} y^{-((s_3-\tilde{s}_3)/(\tilde{c}-c))} a^{-2^{\lambda_1}} a_0^{-1})^{\alpha_2} a_0^{\beta_2}$, this implies it is a contradiction of the strong RSA assumption.

Secondly, in case that $((\tilde{c}-c)/\delta_1)=((\tilde{c}-c)/\delta_2)=((\tilde{c}-c)/\delta_3)=((\tilde{c}-c)/\delta_4)=1$, we can claim that the knowledge extractor is also able to recover the group certificate (A_i,e_i) and the corresponding membership secret x_i as follows.

- Since $\tilde{c}-c=\delta_4=\gcd(s_4-\tilde{s}_4, \tilde{c}-c)$, we have $s_4-\tilde{s}_4=\tau_4(\tilde{c}-c)$, where $\tau_4 \in \mathbb{Z}$. In addition, since we get $r_4 \equiv s_4+c\omega \equiv \tilde{s}_4+\tilde{c}\omega \; (\bmod \varphi(n))$ due to $d_3 \equiv g^{r_4} \equiv g^{s_4+c\omega} \equiv g^{\tilde{s}_4+\tilde{c}\omega} (\bmod n)$, we have $\tau_4 \equiv \omega (\bmod \varphi(n))$ and thus can obtain

$$A_i = \frac{T_1}{y^{\tau_4}} \bmod n.$$

- Since $\tilde{c}-c=\delta_1=\gcd(s_1-\tilde{s}_1, \tilde{c}-c)$, we have $s_1-\tilde{s}_1=\tau_1(\tilde{c}-c)$, where $\tau_1 \in \mathbb{Z}$. In addition, since we get $r_1 \equiv s_1+c(e_i-2^{\gamma_1}) \equiv \tilde{s}_1+\tilde{c}(e_i-2^{\gamma_1}) (\bmod \varphi(n))$ due to $d_4 \equiv g^{r_1}h^{r_4} \equiv g^{s_1+c(e_i-2^{\gamma_1})}h^{r_4} \equiv g^{\tilde{s}_1+\tilde{c}(e_i-2^{\gamma_1})}h^{r_4} (\bmod n)$, we can find

$$e_i \equiv 2^{\gamma_1}+\tau_1 (\bmod \varphi(n)).$$

- Since $\tilde{c}-c=\delta_3=\gcd(s_3-\tilde{s}_3, \tilde{c}-c)$, we have $s_3-\tilde{s}_3=\tau_3(\tilde{c}-c)$, where $\tau_3 \in \mathbb{Z}$. In addition, since we get $r_3 \equiv s_3+ce_i\omega \equiv \tilde{s}_3+\tilde{c}e_i\omega (\bmod \varphi(n))$ due to $d_2 \equiv T_2^{r_1}/g^{r_3} \equiv T_2^{r_1}/g^{s_3+ce_i\omega} \equiv T_2^{r_1}/g^{\tilde{s}_3+\tilde{c}e_i\omega} (\bmod n)$, we can have $\tau_3 \equiv e_i\omega (\bmod \varphi(n))$.
- Since $\tilde{c}-c=\delta_2=\gcd(s_2-\tilde{s}_2, \tilde{c}-c)$, we have $s_2-\tilde{s}_2=\tau_2(\tilde{c}-c)$, where $\tau_2 \in \mathbb{Z}$. In addition, since we get $r_2 \equiv s_2+c(x_i-2^{\lambda_1}) \equiv \tilde{s}_2+\tilde{c}(x_i-2^{\lambda_1}) (\bmod \varphi(n))$ due to $d_1 \equiv T_1^{r_1}/(a^{r_2}y^{r_3}) \equiv T_1^{r_1}/(a^{s_2+c(x_i-2^{\lambda_1})}y^{r_3}) \equiv T_1^{r_1}/(a^{\tilde{s}_2+\tilde{c}(x_i-2^{\lambda_1})}y^{r_3}) (\bmod n)$, we can recover

$$x_i \equiv 2^{\lambda_1}+\tau_2 (\bmod \varphi(n)),$$

Based on that the knowledge extractor can compute the certificate and the corresponding secret, it is sufficient to say that this scheme is a proof of knowledge (Ateniese et al., 2000; Camenisch and Michels, 1998a,b). □

Lemma 2. *Under the strong RSA assumption, a group certificate* $[A_i = (a^{x_i}a_0)^{1/e_i} \bmod n, e_i]$ *with* $x_i \in \Lambda$ *and* $e_i \in \Gamma$ *can be generated only by the group manager provided that the number K of certificates the group manager issues is polynomially bounded.*

Proof. Let $\mathcal{M}$ be an attacker that is allowed to adaptively run the Join procedure of ACJT and thereby obtain group certificates $[A_j = (a^{x_j}a_0)^{1/e_j} \bmod n, e_j]$, $j = 1, \ldots, K$. Our task is now to show that if $\mathcal{M}$ outputs a tuple $(\hat{x}; [\hat{A}, \hat{e}])$, with $\hat{x} \in \Lambda, \hat{e} \in \Gamma$, $\hat{A} = (a^{\hat{x}}a_0)^{1/\hat{e}} \bmod n$, and $(\hat{x}, \hat{e}) \neq (x_j, e_j)$ for all $1 \leq j \leq K$ with non-negligible probability, then the strong RSA assumption does not hold.

Given a pair (n, z), we repeatedly play a random one of the following two games with $\mathcal{M}$ and hope to calculate a pair $(u, e) \in \mathbb{Z}_n^* \times \mathbb{Z} > 1$ satisfying $u^e \equiv z (\bmod n)$ from $\mathcal{M}$'s answers.

The first game **G1** goes as follows:

(1) Select $x_1, \ldots, x_K \in \Lambda$ and $e_1, \ldots, e_K \in \Gamma$.
(2) Set $a = z^{\prod_{1 \leq \ell \leq K} e_K} \bmod n = z^{e_1 e_2 \ldots e_K} \bmod n$.
(3) Choose $r \in_R \Lambda$ and set $a_0 = a^r \bmod n$.
(4) For all $1 \leq i \leq K$, compute $A_i = z^{(x_i + r)\prod_{1 \leq \ell \leq K, \ell \neq i} e_\ell} \bmod n$.
For example, $A_1 = z^{(x_1+r)e_2 e_3 \ldots e_K} \bmod n$, $A_2 = z^{(x_2+r)e_1 e_3 \ldots e_K} \bmod n, \ldots$, and $A_K = z^{(x_K+r)e_1 e_2 \ldots e_{K-1}} \bmod n$.
(5) Select $g, h \in_R QR(n)$, $x \in \{0, \ldots, n^2\}$ and set $y = g^x \bmod n$.
(6) Run the Join protocol K times with $\mathcal{M}$ on input (n, a, a_0, y, g, h) with the prepared x_i, e_i, and A_i for $1 \leq i \leq K$.
(7) After these K registration protocols are done, $\mathcal{M}$ outputs $(\hat{x}; [\hat{A}, \hat{e}])$ with $\hat{x} \in \Lambda, \hat{e} \in \Gamma$, and $\hat{A} = (a^{\hat{x}}a_0)^{1/\hat{e}} \bmod n$.
(8) If $\gcd(\hat{e}, e_j) \neq 1$ for all $1 \leq j \leq K$ then output $\perp$ and quit. Otherwise, let $\tilde{e} := (\hat{x} + r)\prod_{1 \leq \ell \leq K} e_\ell$ (note that $\hat{A} \equiv z^{\tilde{e}} (\bmod n)$). Because $\gcd(\hat{e}, e_j) = 1$ for all $1 \leq j \leq K$, we have $\gcd(\hat{e}, \tilde{e}) = \gcd(\hat{e}, (\hat{x} + r))$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ s.t. $\alpha\hat{e} + \beta\tilde{e} = \gcd(\hat{e}, (\hat{x} + r))$. Therefore, letting $u := z^\alpha \hat{A}^\beta \bmod n$ and $e := \hat{e}/\gcd(\hat{e}, (\hat{x} + r)) > 1$ because $\hat{e} > (\hat{x} + r)$, we have $u^e \equiv z (\bmod n)$. Output (u, e).

The previous game is only successful if $\mathcal{M}$ returns a new certificate $[\hat{A}, \hat{e}]$ with $\gcd(\hat{e}, e_j) = 1$ for all $1 \leq j \leq K$.

We now present another game that solves the strong RSA problem in the other case when $\gcd(\hat{e}, e_j) \neq 1$ for all $1 \leq j \leq K$. Note that $\gcd(\hat{e}, e_j) \neq 1$ means $\gcd(\hat{e}, e_j) = e_j$ because e_j is prime.

The second game **G2** goes as follows:

(1) Select $x_1, \ldots, x_K \in \Lambda$ and $e_1, \ldots, e_K \in \Gamma$.
(2) Choose $j \in_R \{1, \ldots, K\}$ and set $a \equiv z^{\prod_{1 \leq \ell \leq K, \ell \neq j} e_\ell} \equiv z^{e_1 \ldots e_{j-1} e_{j+1} \ldots e_K} (\bmod n)$.
(3) Choose $r \in_R \Lambda$ and set $A_j = a^r \bmod n$ and $a_0 = A_j^{e_j}/a^{x_j} \bmod n$.
(4) For all $1 \leq i \leq K$, $i \neq j$, compute $A_i = z^{(x_i + e_j r - x_j)\prod_{1 \leq \ell \leq K, \ell \neq i,j} e_\ell} \bmod n$.
If we choose $j = 7$ then

$$A_1 = z^{(x_1 + e_7 r - x_7)\prod_{1 \leq \ell \leq K, \ell \neq 1,7} e_\ell} \equiv z^{(x_1 + e_7 r - x_7) e_2 \ldots e_6 e_8 \ldots e_K} (\bmod n),$$
$$A_2 = z^{(x_2 + e_7 r - x_7)\prod_{1 \leq \ell \leq K, \ell \neq 2,7} e_\ell} \equiv z^{(x_2 + e_7 r - x_7) e_1 e_3 \ldots e_6 e_8 \ldots e_K} (\bmod n),$$
$$\vdots$$
$$A_K = z^{(x_K + e_7 r - x_7)\prod_{1 \leq \ell \leq K, \ell \neq K,7} e_\ell} \equiv z^{(x_K + e_7 r - x_7) e_1 \ldots e_6 e_8 \ldots e_{K-1}} (\bmod n).$$

(5) Select $g, h \in_R QR(n)$, $x \in \{0, \ldots, n^2\}$ and set $y = g^x \bmod n$.
(6) Run the Join protocol K times with $\mathcal{M}$ on input (n, a, a_0, y, g, h) with the prepared x_i, e_i, and A_i for $1 \leq i \leq K$.
(7) After these K registration protocols are done, $\mathcal{M}$ outputs $(\hat{x}; [\hat{A}, \hat{e}])$ with $\hat{x} \in \Lambda, \hat{e} \in \Gamma$, and $\hat{A} = (a^{\hat{x}}a_0)^{1/\hat{e}} \bmod n$.
(8) If $\gcd(\hat{e}, e_j) \neq e_j$ output $\perp$ and quit. Otherwise, we have $\hat{e} = te_j$ for some t and can define $Z := \hat{A}^t/A_j \bmod n$ if $\hat{x} \geq x_j$ and $Z := A_j/\hat{A}^t \bmod n$ otherwise. Hence, $Z \equiv (a^{|\hat{x} - x_j|})^{1/e_j} \equiv (z^{|\tilde{e}|})^{1/e_j} (\bmod n)$ with $\tilde{e} := (\hat{x} - x_j)\prod_{1 \leq \ell \leq K; \ell \neq j} e_\ell$. Because $\gcd(e_j, \prod_{1 \leq \ell \leq K; \ell \neq j} e_\ell) = 1$, it follows that $\gcd(e_j, |\tilde{e}|) = \gcd(e_j, |\hat{x} - x_j|)$. Hence, there exist $\alpha, \beta \in \mathbb{Z}$ s.t. $\alpha e_j + \beta|\tilde{e}| = \gcd(e_j, |\hat{x} - x_j|)$. So, letting $u := z^\alpha Z^\beta \bmod n$ and $e := e_j/\gcd(e_j, |\hat{x} - x_j|) > 1$ because $e_j > |\hat{x} - x_j|$, we have $u^e \equiv z (\bmod n)$. Output (u, e).

Consequently, by playing randomly one of the Game **G1** or **G2** until the result is not $\perp$, an attacker getting access to machine $\mathcal{M}$ can solve the strong RSA problem in expected running-time polynomial in K. □

REFERENCES

An JH, Dodis Y, Radbin T. On the security of joint signature and encryption, Advanced in cryptology – EUROCRYPT'2002. In: Lecturer notes in computer science, vol. 2332. Springer Verlag; 2002. p. 83–107.

Ateniese G, Camenisch J, Joye M, Tsudik G. A practical and provably secure coalition-resistant group signature scheme, Advanced in cryptology – CRYPTO'2000. In: Lecturer notes in computer science, vol. 1880. Springer Verlag; 2000. p. 255–70.

Baek J, Steinfeld R, Zheng Y. Formal proofs for the security of signcryption, Public key cryptography – PKC'2002. In: Lecturer notes in computer science, vol. 2274. Springer Verlag; 2002. p. 80–98.

Barić N, Pfitzmann B. Collision-free accumulators and fail-stop signature schemes without trees, Advances in cryptology – EUROCRYPT'97. In: Lecturer notes in computer science, vol. 1233. Springer Verlag; 1997. p. 480–94.

Bellare M, Rogaway P. Random oracles and practical: a paradigm for designing efficient protocols. In: First ACM conference on computer and communication security. ACM Press; 1993. p. 62–73.

Boneh D. The decision Diffie–Hellman problem. In: Proceedings of the third algorithmic number theory symposium. Lecturer notes in computer science, vol. 1423. Springer Verlag; 1998. p. 48–63.

Boyen X. Multipurpose identity-based signcryption: a Swiss army knife for identity-based cryptography, Advanced in cryptology – CRYPTO'2003. In: Lecturer notes in computer science, vol. 2729. Springer Verlag; 2003. p. 383–99.

Bresson E, Chevassut O, Essiari A, Pointcheval D. Mutual authentication and group key agreement for low-power mobile devices. In: Proceedings of the fifth IFIP-TC6 international conference on mobile and wireless communications networks. World Scientific Publishing; 2003. p. 59–62.

Camenisch J, Michels M. A group signatures scheme with improved efficiency, Advances in cryptography – ASIACRYPT'98. In: Lecturer notes in computer science, vol. 1514. Springer Verlag; 1998a. p. 160–74.

Camenisch J, Michels M. A group signatures scheme based on an RSA-variant, Technical report RS-98–27, BRICS. University of Aarhus; 1998b.

Camenisch J, Stadler M. Efficient group signature schemes for large groups, Advances in cryptography – CRYPTO'97.

In: Lecturer notes in computer science, vol. 1294. Springer Verlag; 1997. p. 410–24.
Fujisaki E, Okamoto T. Statistical zero knowledge protocols to prove modular polynomial relations, Advanced in cryptology – CRYPTO'97. In: Lecturer notes in computer science, vol. 1297. Springer Verlag; 1997. p. 16–30.
Kwak D, Moon S. Efficient distributed signcryption scheme as group signcryption, First applied cryptography and network security – ACNS'03. In: Lecturer notes in computer science, vol. 2846. Springer-Verlag; 2003. p. 403–17.
Mu Y, Varadharajan V. Distributed signcryption. In: Advanced in cryptology – INDOCRYPT'2000 proceedings. Lecturer notes in computer science, vol. 1977. Springer Verlag; 2000. p. 155–64.
Mu Y, Varadharajan V, Nguyen KQ. Delegated decryption, Cryptography and coding'99. In: Lecturer notes in computer science, vol. 1746. Springer Verlag; 1999. p. 258–69.
Nakanishi T, Sugiyama Y. Anonymous statistical survey of attributes. In: Sixth Australasian conference on information security and privacy – ACISP 2001. Lecturer notes in computer science, vol. 2119. Springer Verlag; 2001. p. 460–73.
Nakanishi T, Tao M, Sugiyama Y. A group signature scheme committing the group, Information and communications security – ICICS 2002. In: Lecturer notes in computer science, vol. 2513. Springer Verlag; 2002. p. 73–84.
Nam J, Kim S, Won D. Attacks on Bresson–Chevassut–Essiari–Pointcheval's group key agreement scheme for low-power mobile devices. In: Cryptology ePrint archive, Report 2004/251; 2004.
Steinfeld R, Zheng Y. A signcryption scheme based on integer factorization, Information security workshop – ISW'00. In: Lecturer notes in computer science, vol. 1975. Springer Verlag; 2000. p. 308–22.
Wang G, Deng RH, Kwak D, Moon S. Security analysis of two signcryption scheme. In: Information security conference – ISC 2004. Lecturer notes in computer science, vol. 3225. Springer Verlag; 2004. p. 123–33.
Zheng Y. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption). In: Advanced in cryptology – CRYPTO'97 proceedings. Lecturer notes in computer science, vol. 1294. Springer Verlag; 1997. p. 165–79.
Zheng Y. Signcryption and its application in efficient public key solutions, Information security workshop – ISW'97. In: Lecturer notes in computer science, vol. 1396. Springer Verlag; 1997. p. 291–312.

DongJin Kwak is currently a research scientist with KT(Korea Telecom) Future Advanced Technology Laboratory, Korea. He received his B.E., M.E and Ph.D. degrees in School of Electrical Engineering and Computer Science, Kyungpook National University, Korea, in 1998, 2000 and 2005, respectively. He has more than 10 technical publications in the areas of cryptography, information security, and electronic commerce. His main research interests include the analysis, design, and application of digital signature, signcryption, mobile IPv6 security and security protocol etc.

SangJae Moon received his B.E. and M.E. degrees in Electronics from Seoul National University, Korea, in 1972 and 1974, respectively. He received his Ph.D. in Communication Engineering from the University of California, Los Angeles, USA, in 1984. He was working as a consultant of Omnet, Co., USA from 1984 to 1985.

Currently, he is a professor with the School of Electrical Engineering and Computer Science, Kyungpook National University, Korea, and the director of Mobile Network Security Technology Research Center (MSRC). He is also an honorary president of the Korea Institute of Information Security and Cryptology.

His current research interests are the information security in mobile, ubiquitous, and RFID networks including the physical security on smart IC cards. He took part in the Korea Certificate-based Digital Signature Algorithm (KCDSA) Standard project. He has a number of issued patents and more than one hundred technical publications in international journals and conferences in the areas of information security.

Guilin Wang is currently a research scientist with the Institute for Infocomm Research (I2R), Singapore. Before jointing I2R in June 2002, he was an assistant professor at the Institute of Software, Chinese Academy of Sciences. He received his Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, China, in March 2001. He has more than 50 technical publications in the areas of cryptography, information secuity, and electronic commerce. His main research interests include the analysis, design, and application of digital signature, secret sharing, and security protocol etc. Dr. Wang has served as a program committee member or reviewer for a lot of international conferences, workshops and journals. His homepage is <http://www.i2r.a-star.edu.sg/icsd/staff/guilin>.

Robert H. Deng is Professor and Director of SIS Research Center, School of Information Systems, Singapore Management University. Prior to this, he was Principal Scientist and Manager of Infocomm Security Department, Institute for Infocomm Research. He has 23 patents and more than 140 technical publications in international conferences and journals in the areas of computer networks, network security and information security. He has served as general chair, program chair, and program committee member of numerous international conferences. He received the University Outstanding Researcher Award from the National University of Singapore in 1999 and the Lee Kuan Yew Fellow for Research Excellence from the Singapore Management University in 2006. He received his B.Eng from National University of Defense Technology, China, in 1981, his M.Sc and Ph.D. from Illinois Institute of Technology, USA, in 1983 and 1985, respectively.